

**IMPORTANTE:**

- Crear un proyecto con su Apellido y Nombre.
  - Realizar todas las funciones que se indican.
  - Añadir comentarios a su código identificando cada inciso, enumerarlos según corresponda
  - Desarrolle un main() e invoque únicamente a las funciones que compilen correctamente.
- La resolución del mismo es obligatoria y sumará puntos extras en caso de necesitarlos.
- **EL PROGRAMA DEBE COMPILAR SIN ERRORES. DEBE IR VERIFICANDO SI COMPILA BIEN CADA FUNCIÓN REALIZADA. ENTREGUE SOLAMENTE LO QUE COMPILA BIEN.**

Dentro de un sistema de administración de una liga de fútbol, sean las siguientes estructuras de datos y archivo:

Estructura RegistroJugador
<pre>typedef struct {     int idRegistro;     int nroJugador;     char nombreJugador[30];     char apellidoJugador[30];     int clase; // año de nacimiento     char puestoJugador[30]; /**Delantero - Medio - Defensor - Arquero*/     char nombreEquipo[30];     int puntosGanados; } stRegistroJugador;</pre>

Estructura Jugador	Lista de jugadores
<pre>typedef struct {     int nroJugador;     char nombreJugador[30];     char apellidoJugador[30];     int clase;     char puestoJugador[30]; } stJugador;</pre>	<pre>typedef struct {     stJugador j;     struct nodoListaJugador * sig; } nodoListaJugador;</pre>

Estructura Equipo	Arreglo de listas
<pre>typedef struct {     int idEquipo; (generarlo, no está en el archivo)     char nombreEquipo[30];     int puntosGanados; } stEquipo;</pre>	<pre>typedef struct {     stEquipo e;     nodoListaJugador * listaJugadores; } arregloEquipo;</pre>

Archivos entregados (debe utilizar este archivo como único origen de datos del sistema)  
**registroJugador.dat** que contiene 100 jugadores con su correspondiente equipo.

Desarrollar las funciones necesarias para gestionar una Arreglo de Listas. La información se organiza de la siguiente forma:

Un Arreglo de Equipos está compuesto de:

- i. un tipo de dato Equipo,
- ii. un puntero a la raíz de una Lista de jugadores

El sistema tiene que leer el archivo y agregar los jugadores e ir organizándolos en los equipos al que pertenecen.

Obtenido	Valor	Inciso
	15	1. Funciones a desarrollar de la librería de Lista: <ol style="list-style-type: none"> <li>a. Inicializar la lista <b>(1,5 puntos)</b></li> <li>b. Crear un nodo de lista <b>(3,5 puntos)</b></li> <li>c. Insertar en la lista un nuevo nodo de forma recursiva <b>(10 puntos)</b></li> </ol>
	35	2. Funciones para desarrollar del arreglo: <ol style="list-style-type: none"> <li>a. Buscar la posición de un equipo determinado <b>(10 puntos)</b></li> <li>b. Crear un nuevo equipo <b>(10 puntos)</b></li> <li>c. Subprograma: Dar de alta un nuevo jugador en un equipo (invoca a las anteriores a y b) <b>(15 puntos)</b></li> </ol>
	15	3. Hacer una función que lea el archivo e integre todas las funciones del punto 1 y 2 y cargue los 100 jugadores y sus equipos leyendo los registros del archivo <b>registroJugador.dat</b>
	15	4. Realizar una función que muestre todos los jugadores organizados por equipo. <b>Modularizar.</b>
	20	5. Codifique las funciones que necesita realizar para resolver la tarea de buscar <b>el número de un jugador conociendo su equipo, su nombre y apellido</b> . Para desarrollar las funciones, piense qué responsabilidad tiene cada TDA trabajado.  Aclaración: si el número del jugador no se encuentra ingresado (por falta de equipo o falta de nombre y apellido) la función principal debe retornar -1. Caso contrario, retorna el número del jugador.

**Tabla de puntuación:**

Obtenido	10	20	30	40	50	60	70	80	90	100
Nota	1	2	3	4	5	6	7	8	9	10
	Desaprobado					Aprobado				